Computational Aspects of Lattice QCD

or

A “drunkard's walk” through fields of clover.

Bálint Joó
Jefferson Lab, Newport News, VA 23606
bjoo@jlab.org

NCCS Seminar Series, March 2, 2009

Contents

- Introduction to QCD – motivation
- Lattice QCD Method – Monte Carlo & Hybrid Monte Carlo
- Software Details & Performance issues.
- Science Highlights and Summary.

Introduction To QCD

In the Feynman Path Integral formalism, we write a theory as:

"Functional Integral" over all the possible states of the fields

The "action" defining the theory

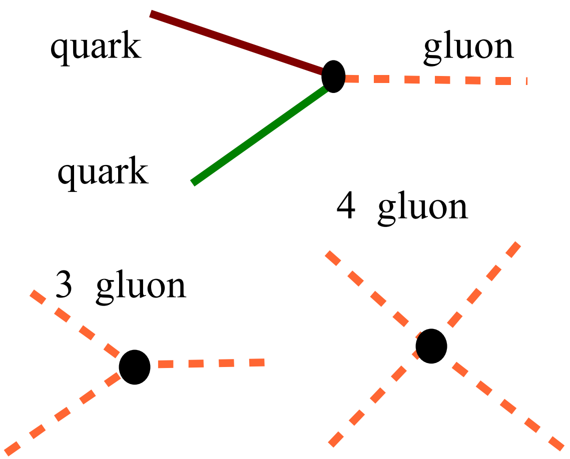
$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \overbrace{\mathcal{D}A \; \mathcal{D}\bar{\psi} \; \mathcal{D}\psi}^{} \; \mathcal{O} \; e^{-S(A,\bar{\psi},\psi)}$$

Expectation value of an observable (eg: particle mass)

Value of observable on a concrete set of fields

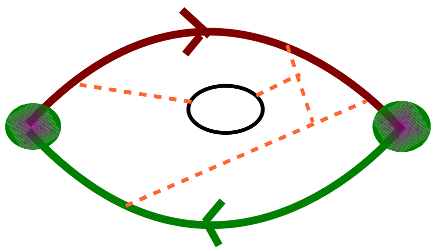
Action enumerates interactions:

$$S = \int dx \; dy \; \bar{\psi}(y) M(A; y, x) \psi(x) - \int dx \; \frac{1}{4} G_a^{\mu\nu}(x) G^a_{\mu\nu}(x)$$

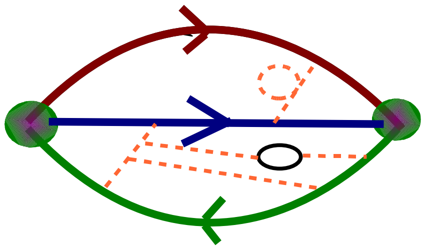


Nature is Colorless

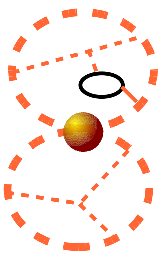
Color charges 'annihilate' at interaction points



We see mesons (2 quarks)



We see Baryons (3 quarks)



Theory predicts 'pure glue' states:
glueballs

Virtual interactions in 'seething vacuum':
quark pairs created from vacuum, destroyed, scattered etc

Important Physics Questions

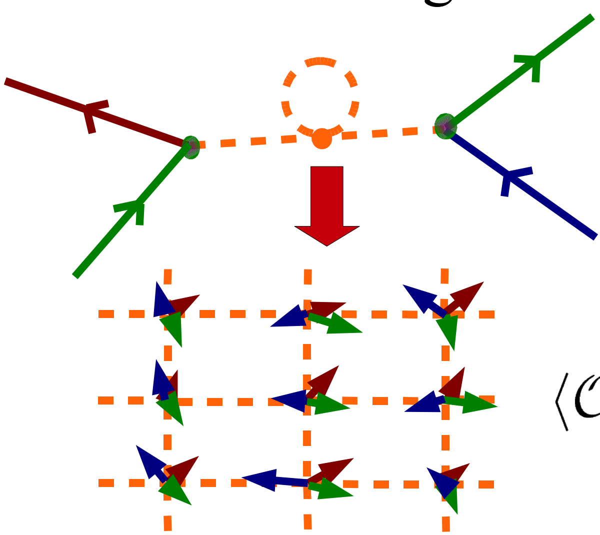
- What are the effective degrees of freedom for low energy nuclear physics?
- What is the role of glue in properties of baryons and mesons ?
- Can QCD explain the spectrum of observed particles?
- Can residual QCD interactions bind together nuclei ?

BUT THERE ARE HURDLES

- Interaction strength of QCD is large
  - perturbation theory fails at low energies
  - need a non-perturbative methodology
- Lattice QCD is the only ab-initio, nonperturbative, model independent method around.

Lattice QCD Prescription

- Move to Euclidean Space, Replace space-time with lattice
- Move from Lie Algebra su(3) to group SU(3) for gluons
- Gluons live on links (Wilson Lines) as SU(3) matrices
- Quarks live on sites as 3-vectors.
- Produce Lattice Versions of the Action
  - derivatives $\rightarrow$ finite differences
  - integrals $\rightarrow$ sums



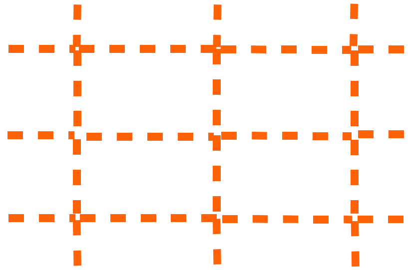
$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \mathcal{D}A \; \mathcal{D}\bar{\psi} \; \mathcal{D}\psi \; \mathcal{O} \; e^{-S(A,\bar{\psi},\psi)}$$

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \prod_{\text{all links}} dU \prod_{\text{all sites}} d[\bar{\psi},\psi] \; \mathcal{O} \; e^{-S(U,\bar{\psi},\psi)}$$

A Statistical Mechanical Analogy

Lattice QCD

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \prod_{\text{links}} dU_i \; \mathcal{O} \; e^{-S(U)}$$



- Configuration: A set of links U
- Probability of Configuration:

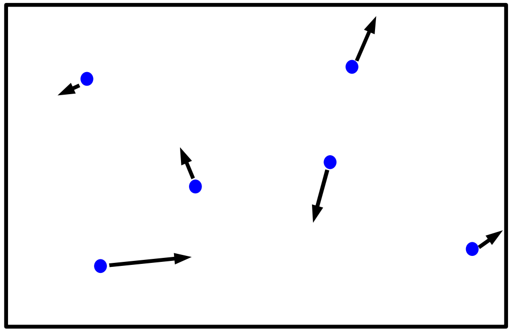
$$P(U) = e^{-S(U)}$$

- Couplings: interaction strength

Dropped the fermions for now, for simplicity

Simulating a Gas

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \prod_{\text{particles}} d\vec{p}_i \; d\vec{q}_i \; \mathcal{O} \; e^{-H(\{\vec{p}_i\},\{\vec{q}_i\})}$$



- Configuration: particle positions and momenta
- Probability of Configuration:

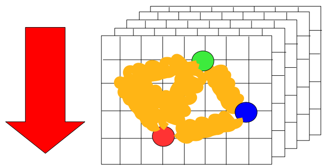
$$P(U) = e^{-H(\{\vec{p}_i\},\{\vec{q}_i\})}$$

- Couplings: E/kT (Boltzmann)

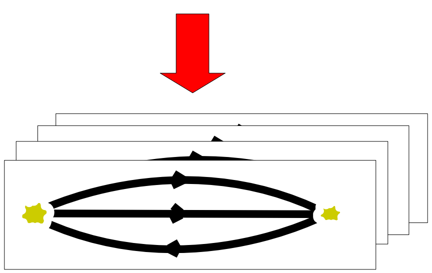
Large Scale LQCD Simulations Today

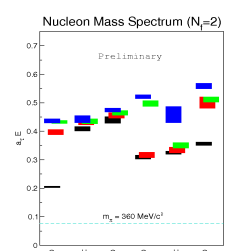
- Stage 1: Generate Configurations
  - via Markov Chain Monte Carlo
  - single chain
  - requires large capability machine
  - discuss this further on

Focus of rest of talk



- Stage 2: Analysis of Configurations
  - soon/now more FLOPS than gauge generation
  - **BUT** task parallelizable (per configuration)
  - each task still numerically intensive
  - efficient (currently) on large capacity clusters or multiple smaller partitions of capability machine





- Stage 3: Extract Physics
  - on workstations, small cluster partitions

Monte Carlo Method

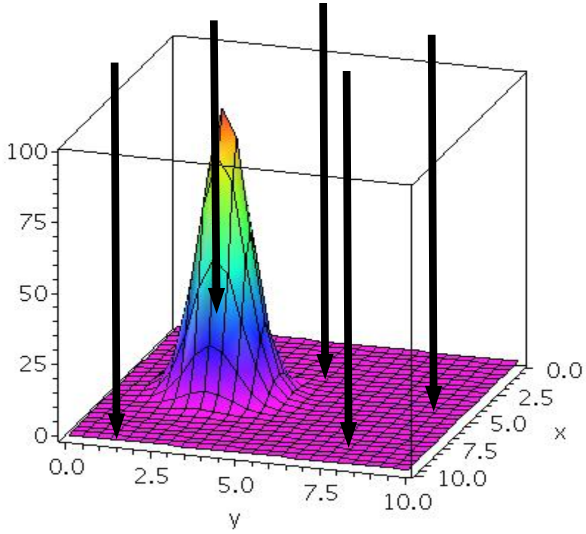
Evaluating the Path Integral:

- There are 4V links. V~16^3x64 - 32^3x256 → 4V = **1M ~ 33M links**
- Direct evaluation unfeasible. Turn to Monte Carlo methods

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \prod_{\text{all links}} dU_i \, \mathcal{O} \; e^{-S(U)} \longrightarrow \bar{O} = \frac{1}{Z} \sum_{\text{configuration}} \mathcal{O}(U) \; P(U)$$

- Basic Monte Carlo Recipe
  - Generate some configurations U
  - Evaluate Observable on each one
  - Form the estimator.

Problem with uniform random sampling:
most configurations have P(U) ~ 0



Importance Sampling

- Pick U, with probability P(U) if possible
- Integral reduces to straight average, errors decrease with statistics

$$\langle \mathcal{O} \rangle = \frac{1}{\mathcal{Z}} \int \prod_{\text{all links}} dU_i \; \mathcal{O} \; e^{-S(U)} \longrightarrow \bar{O} = \frac{1}{N} \sum_{\text{N}} \mathcal{O}(U) \qquad \sigma(\bar{\mathcal{O}}) \propto \frac{1}{\sqrt{N}}$$

Metropolis Method:

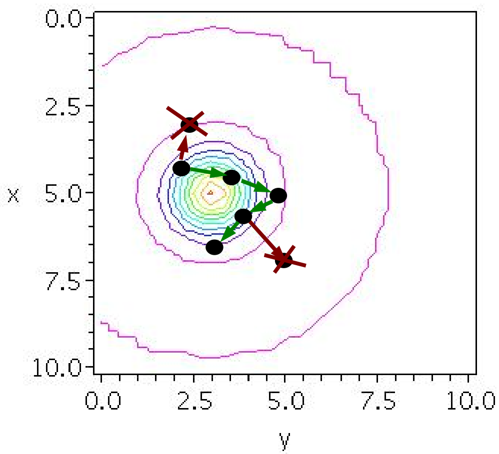
Start from some initial configuration.

Repeat until set of configs. is large enough:

- From config U, pick U' (reversibly)
- Accept with Metropolis probability:

$$P(U' \leftarrow U) = \min\left(1, \frac{e^{-S(U')}}{e^{-S(U)}}\right)$$

- If we reject, next config is U (again)



Generates a Markov Chain of configurations. Errors in observables fall as the number of samples grows

Global Updating

- Imagine changing 'link by link'
- For each change one needs to evaluate the fermion action twice: before and after

$$S_f = \phi^\dagger \left(M^\dagger M \right)^{-1} \phi = \langle \phi | X \rangle$$

where

$$\left(M^\dagger M \right) X = \phi$$

Two Degenerate Flavors of fermion (eg: u & d). Guaranteed
- **Hermitean**
- **Positive Definite**

Use Sparse Krylov Subspace Solver: eg: Conjugate Gradients

Linear system needs to be solved on entire lattice.
- Dimension: ~ O(10M)
- Condition number: O(1-10M)

- 1 Sweep: 2x4V solves, with 4V ~ O(1M-33M) is prohibitive
- Need a Global Update Method

Hybrid Monte Carlo

- Treat Links as 'canonical coordinates' of a Lagrangean
- Find 'canonical momenta'
  - For each link matrix, there is a 'momentum matrix'
  - Configuration space $\rightarrow$ Phase Space
- Define a (fictitious) Hamiltonian

$$H = \frac{1}{2} \sum_{\text{links}} p^2 + S(U)$$

- Momenta come from Gaussian distribution. Generate via a heat bath
- **Simulate extended system:** momentum contributes a constant, which cancels out from observables that are independent of momenta

$$\mathcal{Z} = \int \mathcal{D}U \, \mathcal{D}p \, e^{-H} = \int \mathcal{D}U \, e^{-S} \boxed{\int \mathcal{D}p \, e^{-\frac{1}{2}\sum_{\text{links}} p^2}} = \boxed{C} \int \mathcal{D}U \, e^{-S(U)}$$

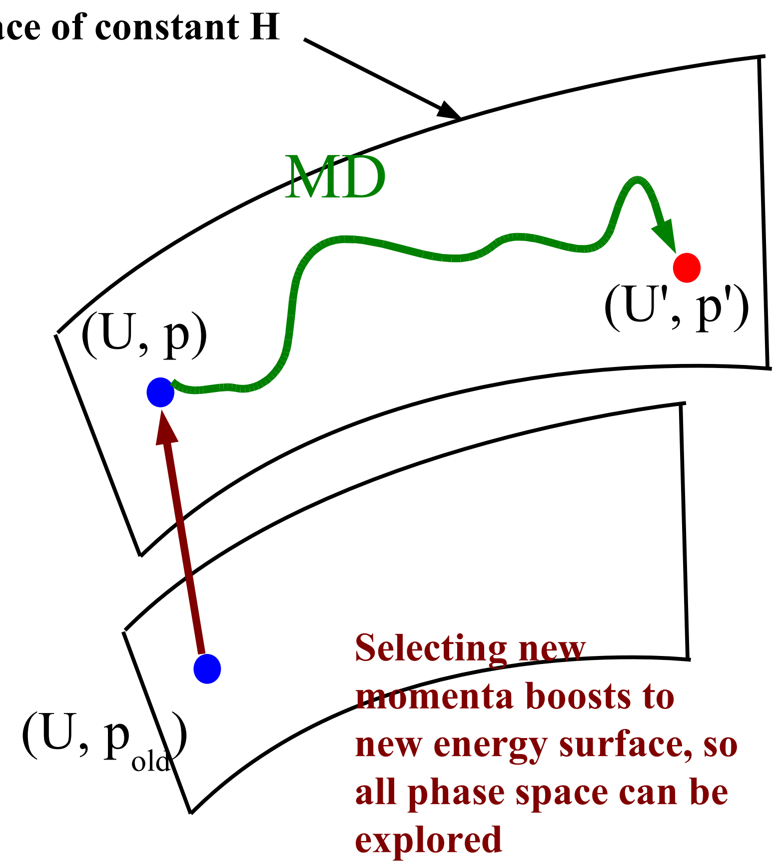
Hybrid Monte Carlo

- **Big Trick: Go from config U to U' doing Hamiltonian Molecular Dynamics in Fictitious Time**

- start from config U
- generate momenta p
- evaluate H(U,p)
- perform MD in fictitious time t
- evaluate H(U', p')
- accept with Metropolis probability

$$P = \min\left(1, e^{-H(U',p')+H(U,p)}\right)$$

- if accepted new config is U', otherwise it is U



MD Conserves Energy
If done exactly P = 1 (always accept)
Otherwise dH depends on the error from the integrator – small.. (?).

Molecular Dynamics

- Reversible and Area Preserving: Reversible combination of symplectic pieces: eg 2nd order leapfrog, 2nd order Omelyan

$$e^{\frac{\delta\tau}{2}\hat{P}} e^{\delta\tau\hat{Q}} e^{\frac{\delta\tau}{2}\hat{P}}$$

$$e^{\lambda\delta\tau\hat{Q}} e^{\frac{\delta\tau}{2}\hat{P}} e^{(1-2\lambda)\delta\tau\hat{Q}} e^{\frac{\delta\tau}{2}\hat{P}} e^{\lambda\ \delta\tau\hat{Q}}$$

- Mulitple Time Scales (Sexton & Weingarten)
  - Split action as $S = S_1 + S_2$

$$S_1, S_2 \longrightarrow \hat{P}_1, \hat{P}_2$$

$$U^{(2)} = e^{\frac{\delta\tau}{2}\hat{P}_2} \left[U\left(\hat{P}_1, \frac{\delta\tau}{N}\right)\right]^N e^{\frac{\delta\tau}{2}\hat{P}_2}$$

- Two time scales: $\delta\tau$ and $\delta\tau$/N, scheme generalizes to more scales
  - Separate action terms with different forces onto different time scales.

Fermion Forces Involve Solvers

2 Flavor Action: $$S = \phi^\dagger \left(M^\dagger M\right)^{-1} \phi$$

$$F = -\phi^\dagger (M^\dagger M)^{-1} \left[\dot{M}^\dagger M + M^\dagger \dot{M}\right] (M^\dagger M)^{-1} \phi$$

$$= -X^\dagger \left[\dot{M}^\dagger M + M^\dagger \dot{M}\right] X$$

$$X = (M^\dagger M)^{-1} \phi$$

Use Conjugate Gradients to compute X

- Need to compute X for every force evaluation.
- For a trajectory with N steps, leapfrog needs N+1 solves
  - much more manageable than than O(V)
  - but still quite expensive numerically

Rational Hybrid Monte Carlo (RHMC)

- For 1 flavor of fermion: M is not guaranteed to be +ve definite. Instead use a square root of the square (or rational approximation of same)

$$S_{1F} = \phi \left(M^\dagger M \right)^{-\frac{1}{2}} \phi \approx \phi^\dagger \left(\boxed{\sum p_i \left[M^\dagger M + q_i \right]^{-1}} \right) \phi$$

$$\approx \sum p_i \langle \phi | X_i \rangle$$

Rational Approximation in Partial Fractional Form. Approximation defined by p_i and q_i

with:

$$\boxed{\left(M^\dagger M + q_i \right) X_i = \phi}$$

- 'Shifted System' with shifts q_i - All X_i are in same Krylov Subspace
- Variants of conjugate gradient can get solutions for all shifts with just 1 solve: so called Multiple-Shift solvers (CG-M)
- Force also needs multiple shift solve.

Solvers

- Energy and Force Calculations:
  - 2 Flavors of Degenerate Quarks: Conjugate Gradients

$$(M^\dagger M)X = \phi$$

  - Or 2 step BiCGStab (though danger of breakdown)

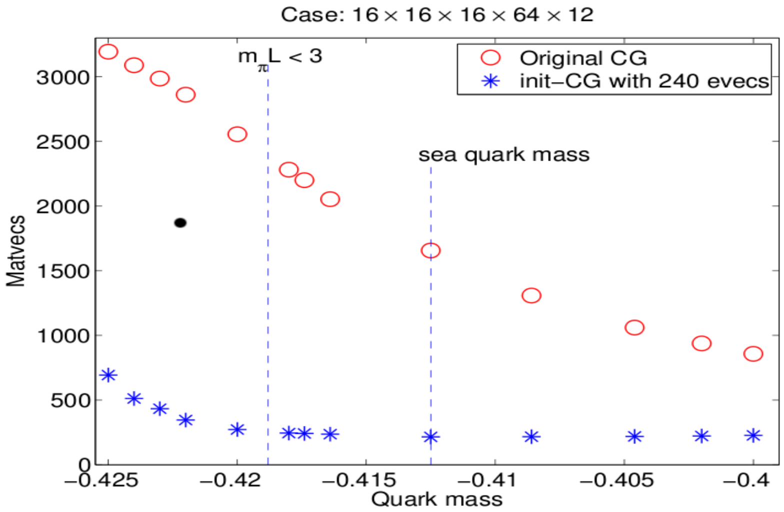
$$M^\dagger Y = \phi \quad MX = Y$$

  - Single Flavor of Degenerate Quark: Shifted CG

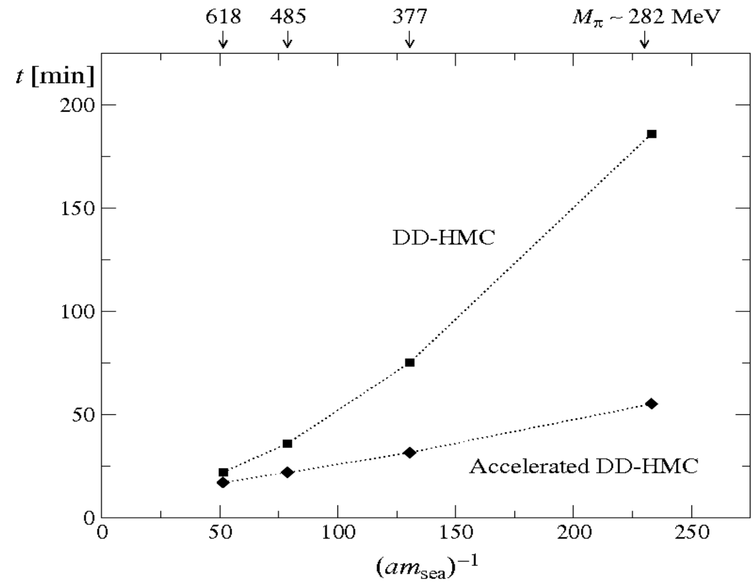
$$(M^\dagger M + q_i)X_i = \phi$$

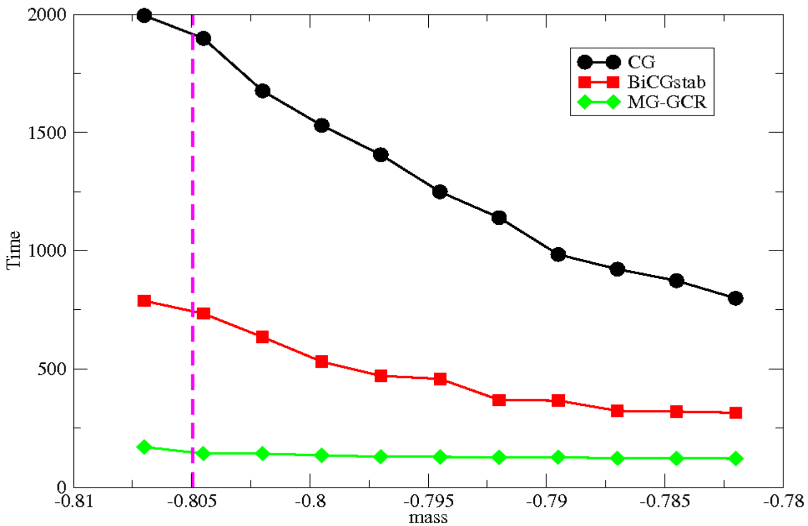
- Critical Slowing Down as quark masses become small
  - number of solver iterations increase as ~ O(1/m)
  - deflation/multigrid techniques can help
  - startup cost for MG/Deflation, but may be still be worth it...

Slowing Down the Slowing Down



EigCG Deflation
(Orginos, Stathopoulos)
arXiv:0707.0131 [hep-lat]





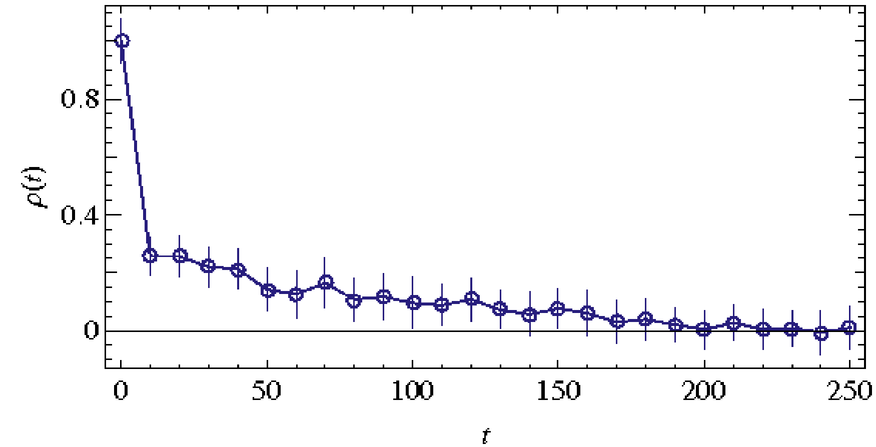
Deflated DD-HMC
(Luscher et al)
arXiv:0710.5417[hep-lat]
JHEP0712:011,2007

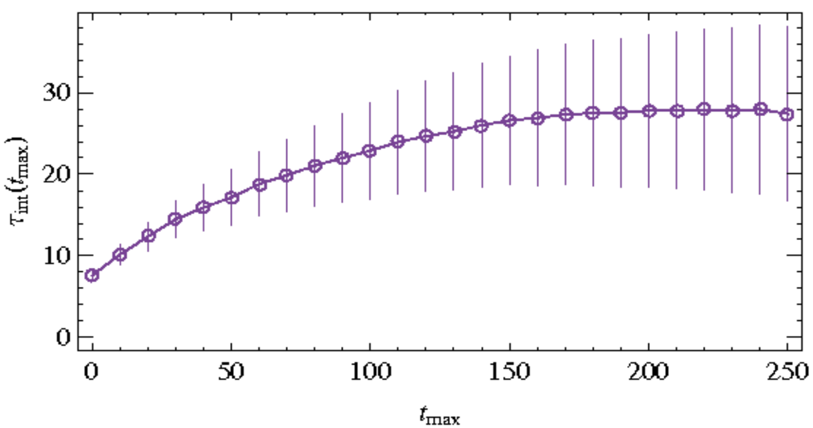
Adaptive Multigrid
(Clark et al)
arXiv:0811.4331 [hep-lat]

Autocorrelations

- Successive configurations may well be correlated
  – in the case of a rejection maximally so...
- For configurations to be independent statistically, they must be separated by the ***autocorrelation time.***
- This enters into the error estimate:

$$\sigma^2(\mathcal{O}) = 2\,\tau_{\mathrm{int}}^{\mathcal{O}}\,\sigma_{\mathrm{naive}}^2(\mathcal{O})$$





- Here, the pion has an autocorrelation time of ~20-30
  – 1000 cfg → 40-60x1000 trj.

Cost of the Monte Carlo Part

- Heuristic Formula, taking into account:
  - volume scaling for MD & Solvers
  - critical slowing down
  - normalized at current simulations

A. Ukawa, HEP Exascale Computing Workshop, 2008

$$C = 0.024 \left(\frac{L^3 T}{(6fm)^4}\right)^{5/4} \left(\frac{135MeV}{m_\pi}\right)^2 \left(\frac{0.1fm}{a}\right)^6 \left(\frac{\#\mathrm{Traj}}{10^4\tau}\right) \ \mathrm{PFLOPSyears}$$

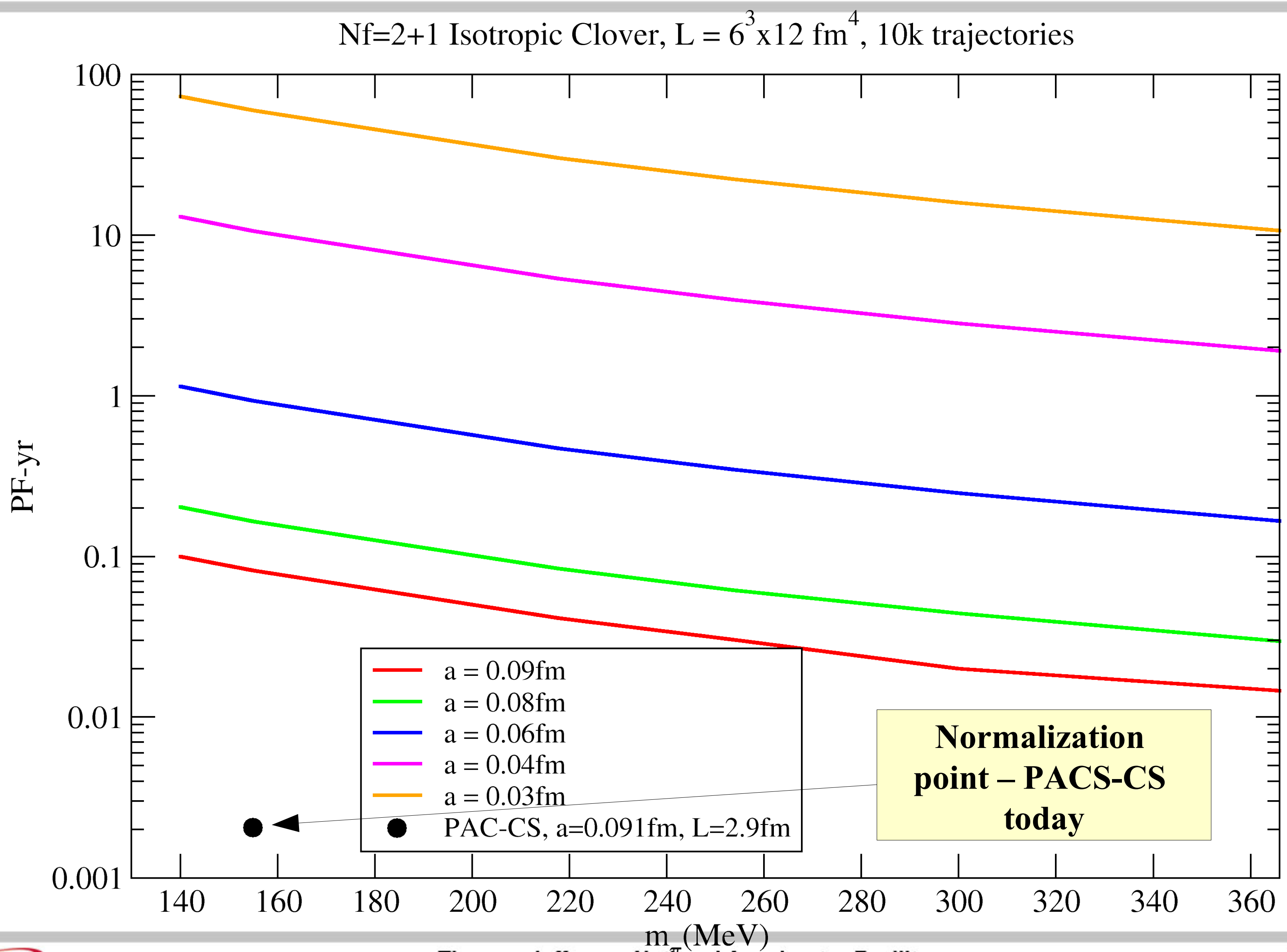
Physical Box Volume

# of HMC trajectories = # indep cfgs x τ

2 powers of a from mass ie $1/(a\, m_\pi)^2 \sim 1/(am_q)$

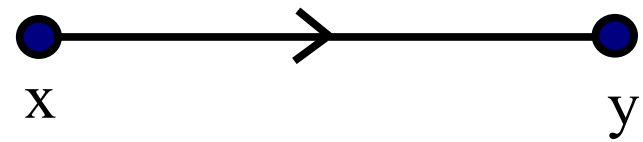
4 powers of a from increase in # lattice points

More on Costs



After the Gauge Generation

Quark Propagator: $G(x,y) = M^{-1}_{x,y} S(x)$



Correlation Functions:

Γ projects onto correct spin-parity quantum numbers

Mesons:

$$C(\vec{p},t) = \sum e^{i\vec{p}.\vec{x}}\ \mathrm{Tr}\ \Gamma\ G^{\dagger}(\vec{x},t;0,0)\ \Gamma\ G(\vec{x},t;0,0)$$

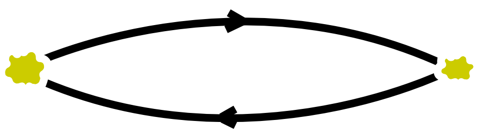
Fourier Transform in space, transforms to Momentum Space.

antiquark

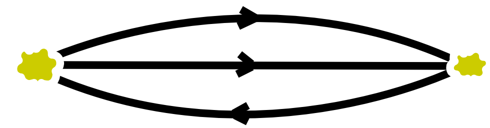
quark

Translation invariance: G(x,0) <=> G(z+x, y)

Meson:



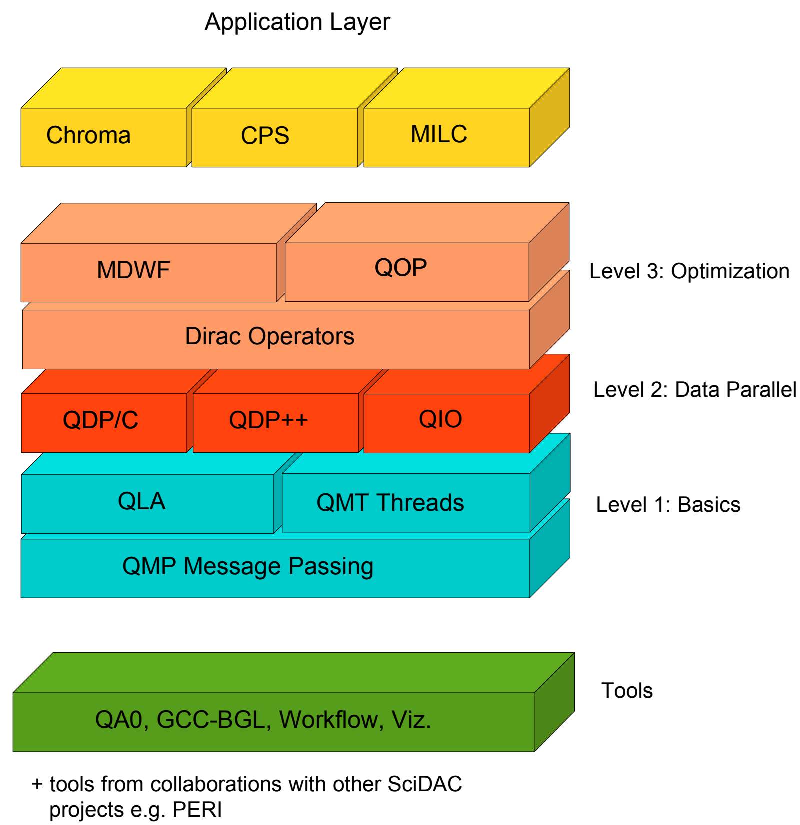
Baryon:



- Measure on each configuration, but only the 'average' is 'physical.
- Baryons also need color antisymmetrization
- Fourier transform fixes definite momenta, but loses volumetric info
  - Not much in the way of pretty visualizations – mostly 2D plots

SciDAC Software for LQCD

- We have developed a wide range of LQCD software under SciDAC.
- Work split into layers
- Level 1: Comms, Threads, Sitewise Linear Algebra
- Level 2: Data Parallel Layer
- Level 3: Optimization layer
  - cut through lower levels for performance
  - solvers, linear operators etc.
- Application Layer:
  - gauge generation (HMC)
  - observable measurement



http://usqcd.fnal.gov The USQCD Web Page
http://usqcd.jlab.org/usqcd-software Software Page

B. Joo, SciDAC 2008, JoP Conf. Ser. 125 (2008) 012066
B. Joo, SciDAC 2007, JoP Conf. Ser. 78 (2007) 012034

The Chroma Stack (for Cray XT)

3rd Party:

GMP

libXML2

Ancilliary Open Source Packages.

Code is freely available but needs coordinated build of 6 modules.

SciDAC LQCD:

Chroma

QCD Library and Application Suite:
Contains HMC algorithm, solvers, MD integrators, observable measuerment codes. Built on QDP++. Uses Level 3 Dslash & Clover Operations on Cray XT3

QDP++

Data Parallel Environment for QCD computations: Hides loops over lattice. Includes QIO for binary IO & XML reader for parameter reading. BLAS like ops optimized with SSE2 compiler intrinsics. C++ with expression templates. Threads via QMT, or OpenMP

QMT

Pthreads based OpenMP like threading library: with fast barriers. Optimized for Barcelona cache coherency

QMP

Message Passing for QCD:
Use reference version built on top of MPI for Cray/XT

MPI + Cray CNL O/S

R. G. Edwards, B. Joo, Nucl. Phys. Proc. Suppl. 140:832, 2005, arXiv:hep-lat/0409003

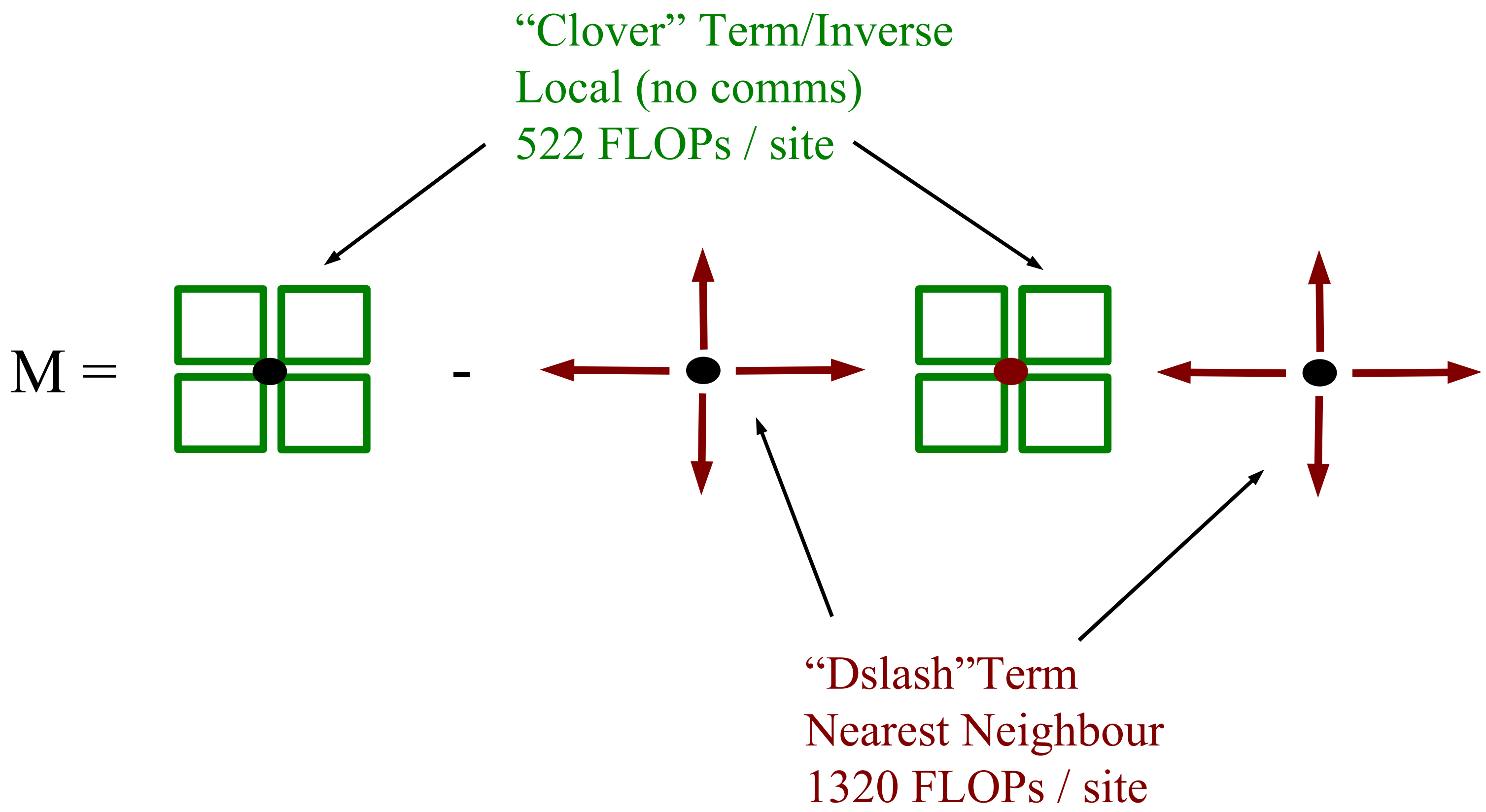
Performance Needs

- Forces, and Gauge Actions need:
  - Level 1 like BLAS like operations on SU(3) matrices
  - Work is local (no comms)
  - Data Parallel approach is very suitable

- Solvers need:
  - Level 1 BLAS like operations on color-vectors
    - all local
  - Global Sums/Inner Products
    - gated by hardware/comms layer
  - Efficient Implementation of the Linear Operator
    - mostly up to us

Our Linear Operator

- We use the so called "Even-Odd Preconditioned Clover" operator.



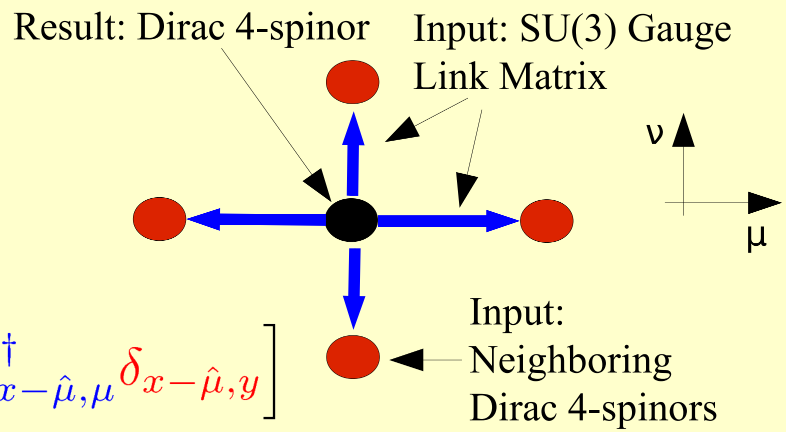
What is the Wilson Dslash?

- It is the lattice discretization of the gauge covariant derivative
- It is a nearest neighbor, finite difference stencil (typically) in 4 dimensions.

Continuum: $\not{D}(x) = \gamma_\mu \partial^\mu - igA_\mu(x)$

Lattice:

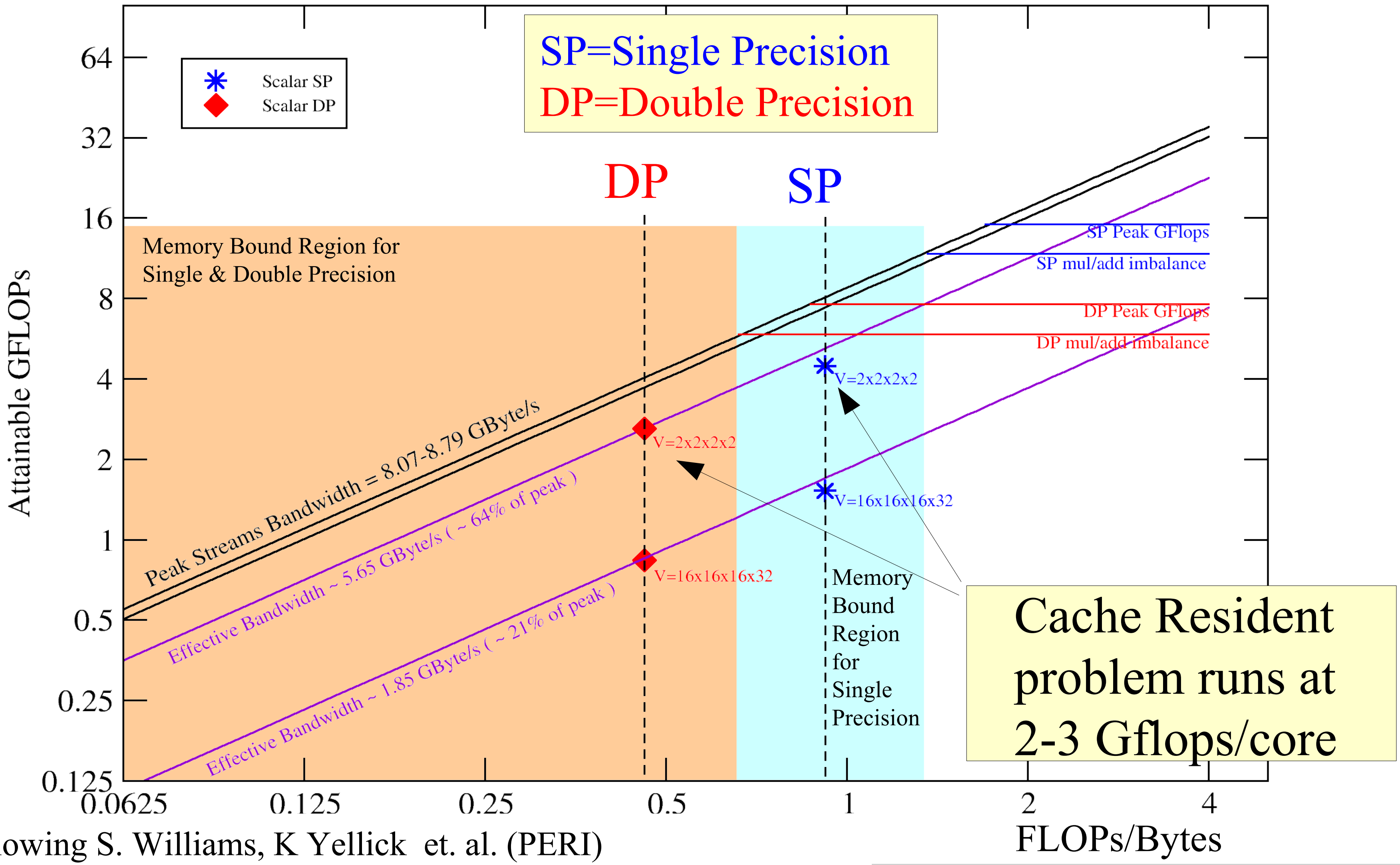
$$\not{D}_{x,y} = \sum_\mu \left[(1-\gamma_\mu) U_{x,\mu} \delta_{x+\hat{\mu},y} + (1+\gamma_\mu) U^\dagger_{x-\hat{\mu},\mu} \delta_{x-\hat{\mu},y} \right]$$



- Compulsory Flops: 1320, FLOPS/Bytes=0.46 (DP), 0.92 (SP)
- Bandwidth bound: Max attainable FLOPS in DP ~ 0.5 B/W

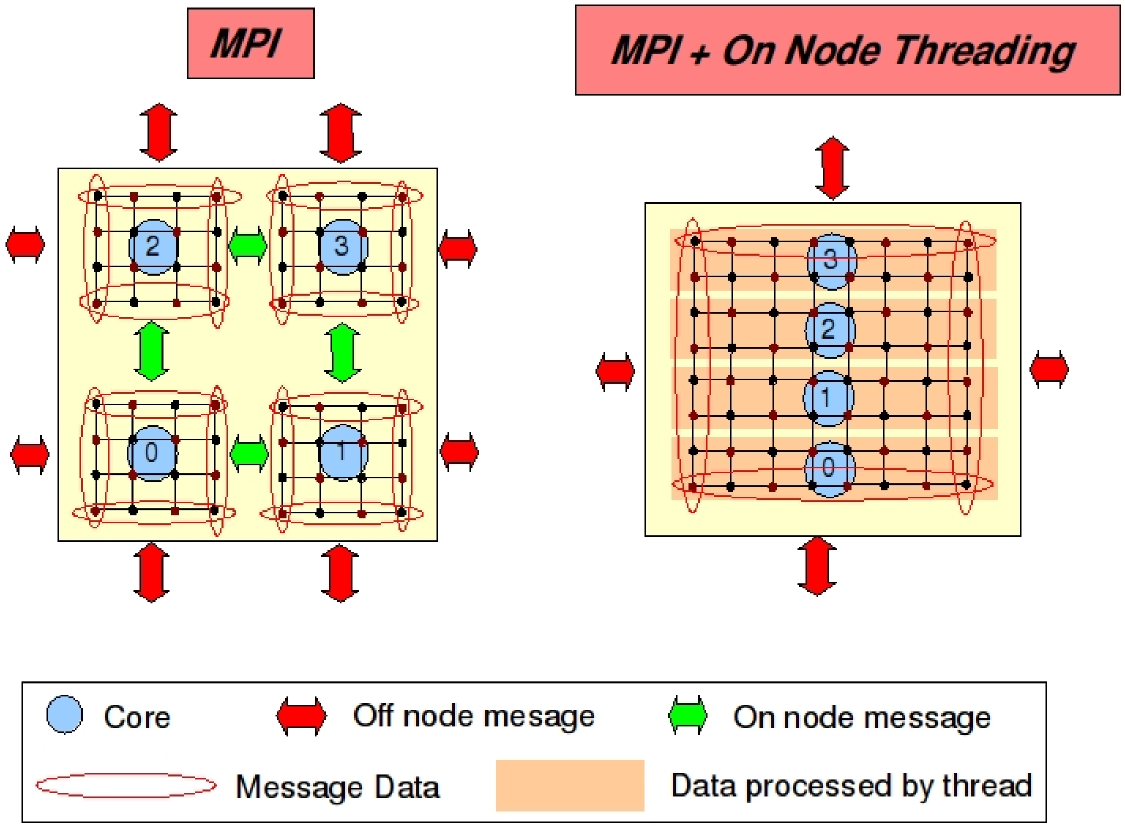
Scalar Dslash – Single core performance

1 Barcelona core@1.9GHz, 15.2GFLOPs peak (SP), 7.6GFLOPs peak (DP)



Following S. Williams, K Yellick et. al. (PERI)

Parallel & Threaded Dslash



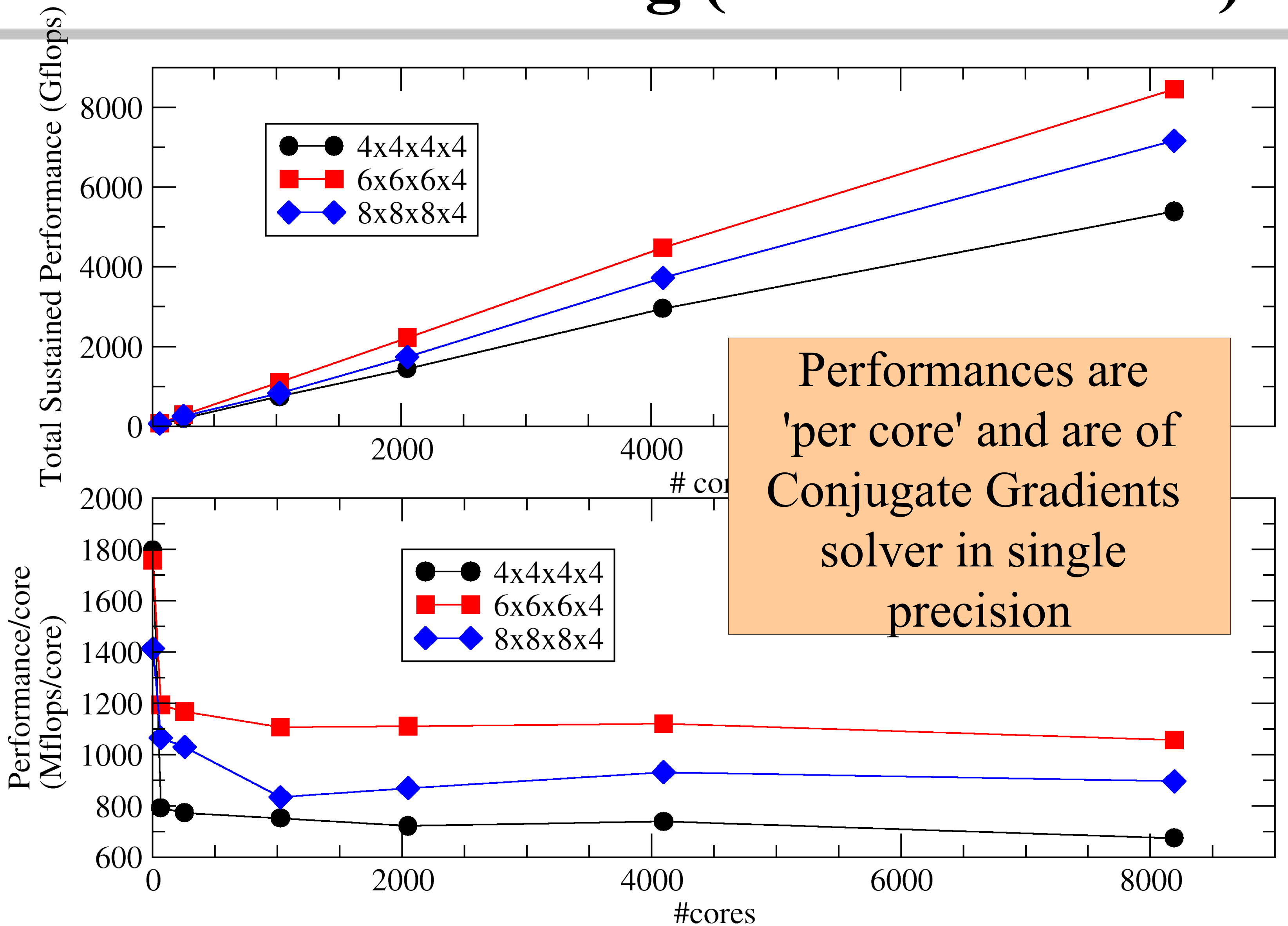
- Potential threading benefits
  - eliminate on node messages (green arrows)
  - coalesce off node message so they are
    - fewer
    - bigger

• Perfect load balancing: All cores/nodes have equal problem size, and message sizes

Expected Scaling

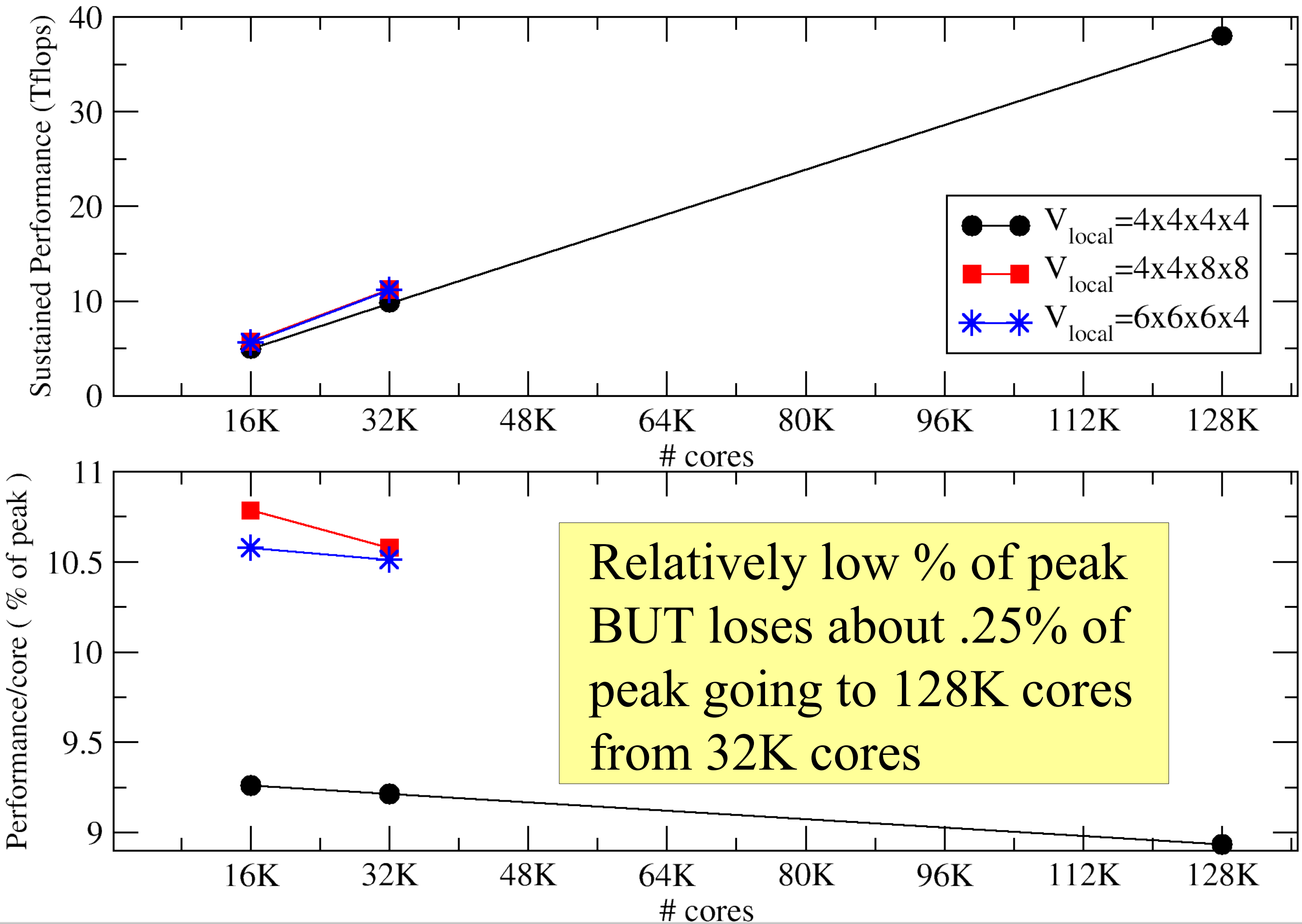
- Our problem is very regular
  - same problem size on each node/core
    - (Body/2) sites x 2 x 1320 FLOPS – Dslash
    - (Body/2) sites x 2 x 522 FLOPS – Clover Term.
  - regular & known communications:
    - 2 x (Face/2) sites x 12 words / direction
  - 1 Global Sum per CG iteration
- Solver performance should weak scale LINEARLY
- Strong scaling should be gated by Surface/Volume
- There really should not be anything irregular
  - No communications imbalance
  - No unexpected messages

Pre CNL Scaling (Dual Core XT3)



Performances are 'per core' and are of Conjugate Gradients solver in single precision

BlueGene/P Scaling



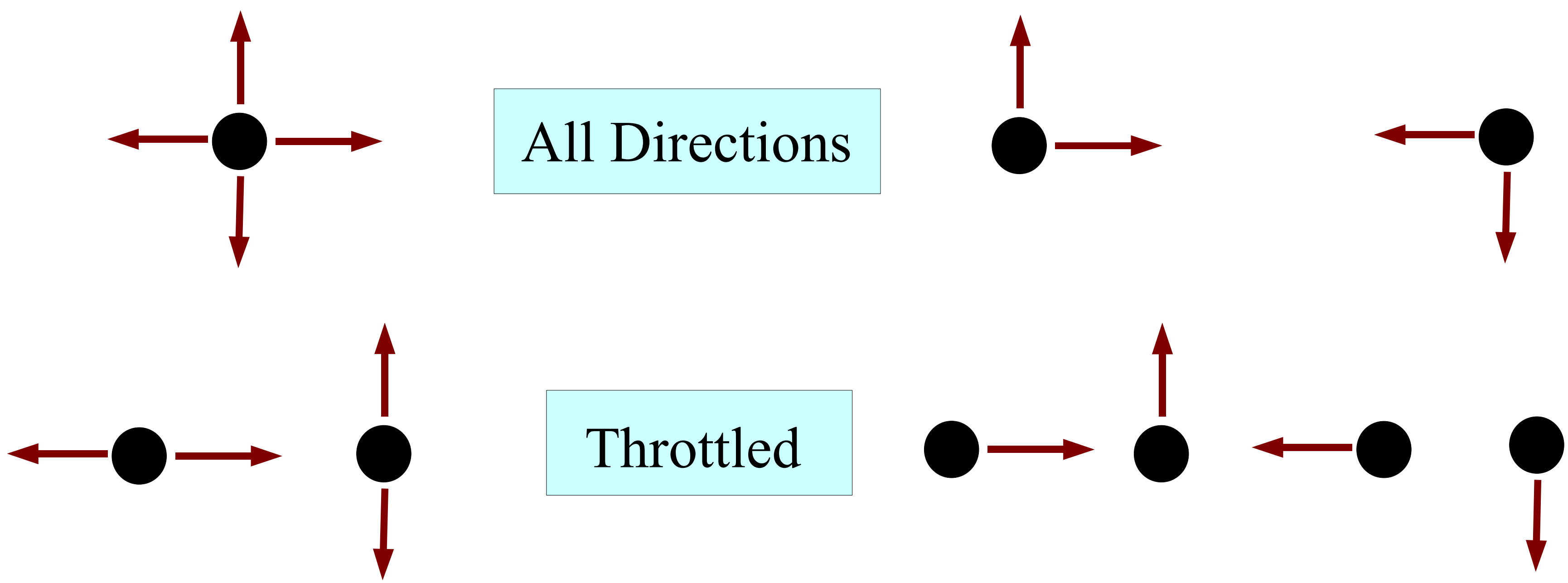
Relatively low % of peak BUT loses about .25% of peak going to 128K cores from 32K cores

Communications Strategies...

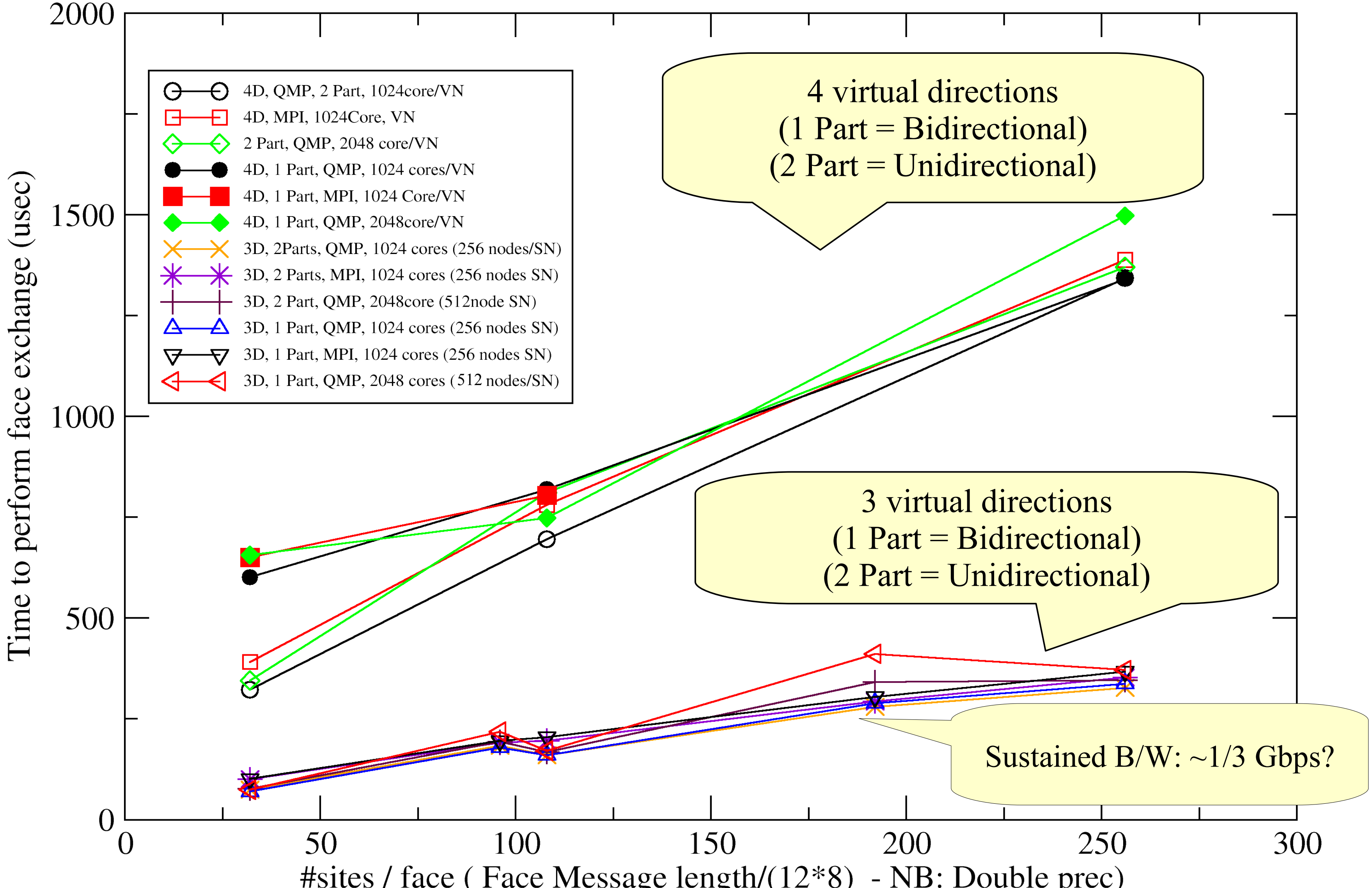
- Can orchestrate a variety of patterns

Bidirectional Sends:
- use full bidirectional bandwidth
- can throttle by limiting # of dimensions in one go...

Unidirectional Sends:
- send forward then backwards
- can throttle on number of directions...

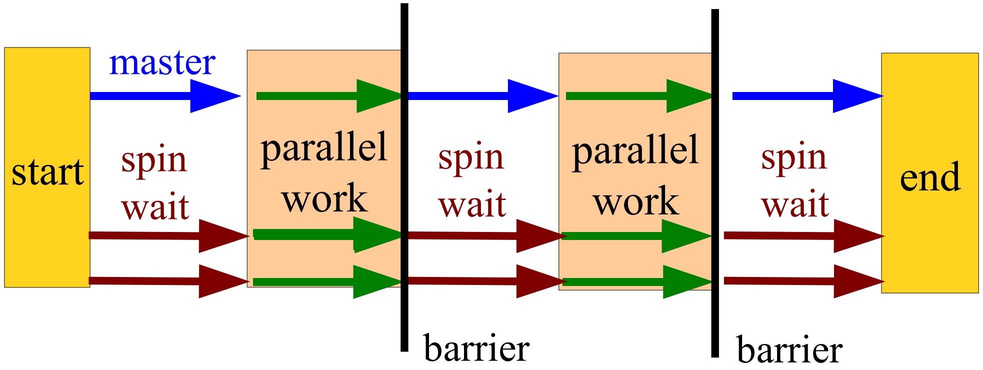


Cray XT4 Comms Characteristics



QMT Highlights

- Threads spawned at startup, joined at end
  - Worker threads spin waiting for work (never idled)
- Master thread shares in parallel work
- Parallel region ended with barrier; called automatically
- Opteron/Intel barrier uses cache coherency for speed
- Like OpenMP #omp_parallel over functions but
  - ThreadArgs and function need to be written for every case.

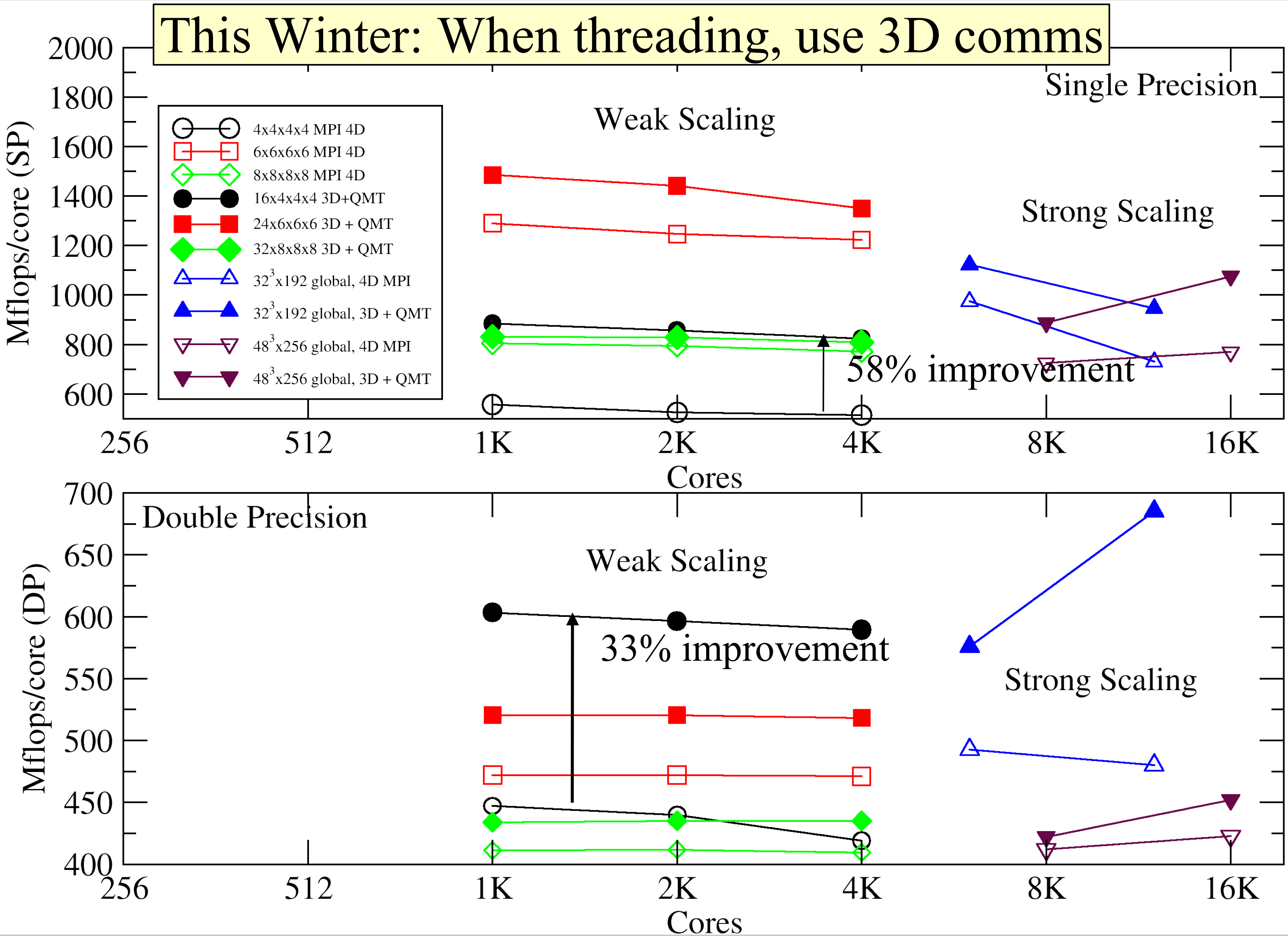



```
#define QUITE_LARGE 10000

typedef struct {
  float *float_array_param;
} ThreadArgs;

void threadedKernel( size_t lo, size_t hi, int id,
                     const void* args)
{
    const ThreadArgs* a = (const ThreadArgs *)args;
    float *fa = a->float_array_param;
    int i;
    for( i=lo; i < hi ; ++i) {   /* DO WORK FOR THREAD */ }
}

int main( int argc, char *argv[] )
{   float my_array[ QUITE_LARGE ];
    ThreadArgs a = { my_array };
    qmt_init();
    qmt_call( threadedKernel, QUITE_LARGE, &a );
    qmt_finalize();
}
```

QMT divides QUITE_LARGE amongst threads to compute lo & hi for each thread. It calls threadedKernel for each thread

a passed straight through to all threads (shared data)

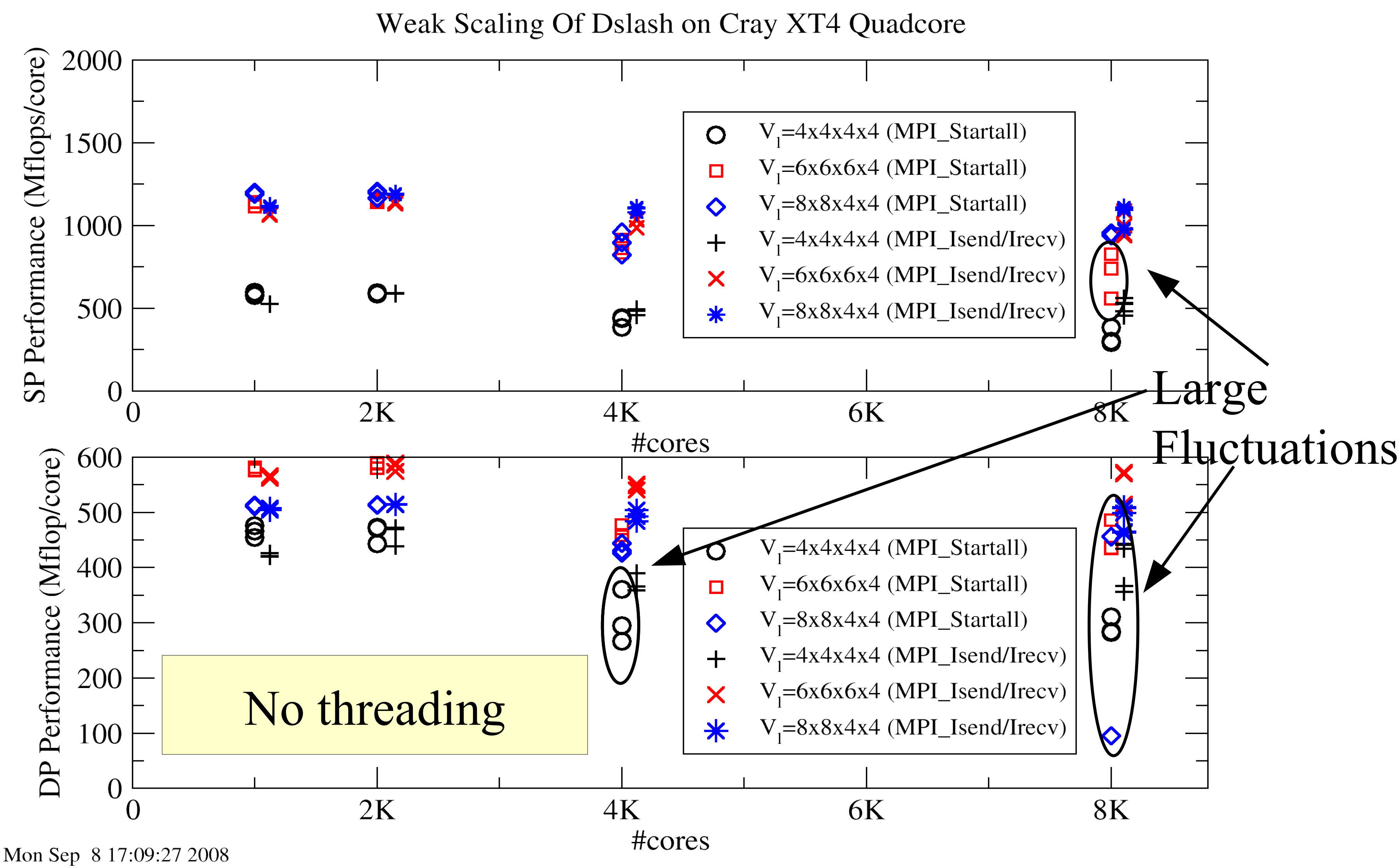
Threading and 3D Comms



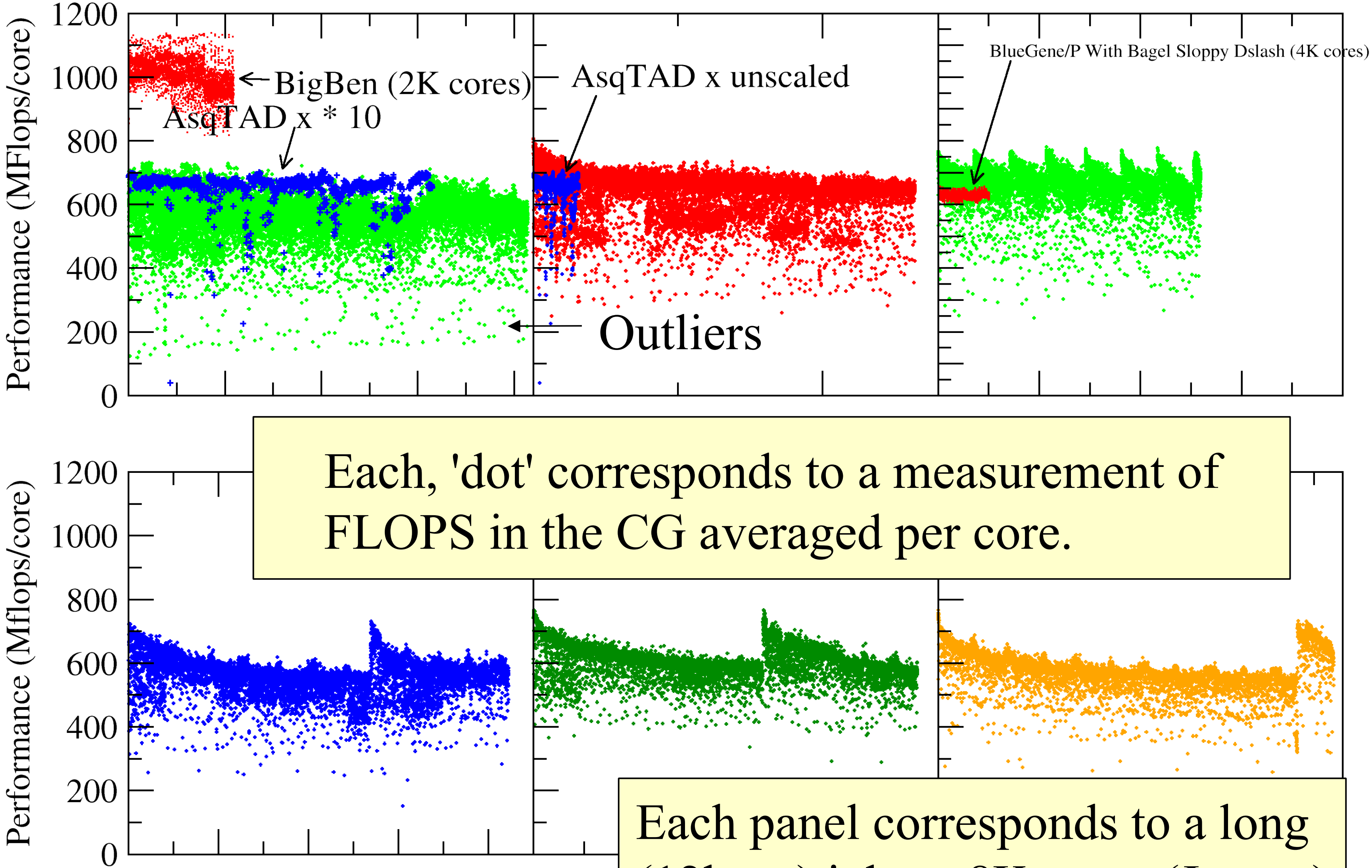
...but nothing is ever easy...

- The QMT Threading seems quite effective
  - especially for smaller local volume (per core) – hard scaling
- But its not all plain sailing:
  - As we moved to larger and larger partitions we began to notice large fluctuations in performance.
    - this happened even without threading...
  - Our threaded code seems to be effective at killing nodes on Kraken XT5
  - Is 550 – 600 Mflops/core really the best we can do in Double Precision?
    - surely, with 6Gbps sustained bidirectional bandwidth in the SeaStars and the good memory bandwidth and cache systems of Opterons we can do better?

Fluctuations in the Dslash (summer 2008)



Fluctuations in Solver Performance



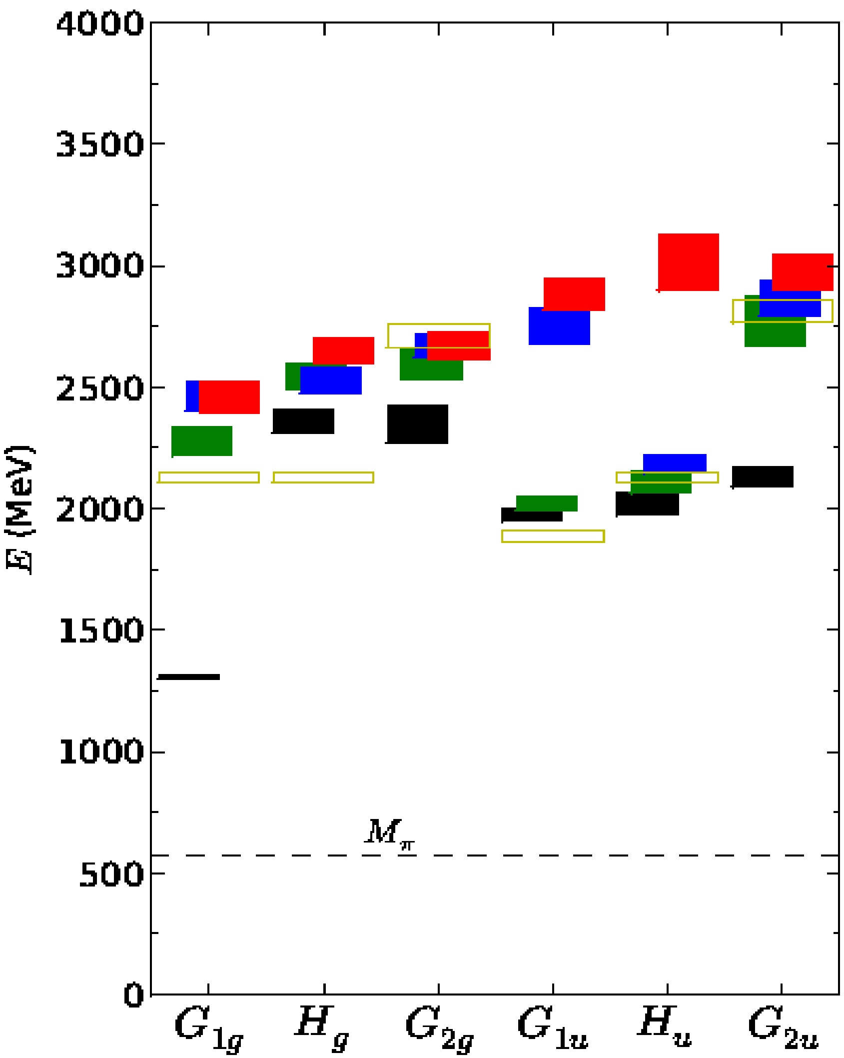
Each, 'dot' corresponds to a measurement of FLOPS in the CG averaged per core.

Each panel corresponds to a long (12hour) job on 8K cores (Jaguar)

Our next steps...

- Understanding and eliminating the fluctuations is an immediate priority (for me)
  - The concrete reason for being here this week.
  - Do others see fluctuating performance like this?
  - Large partitions may be needed for debugging...

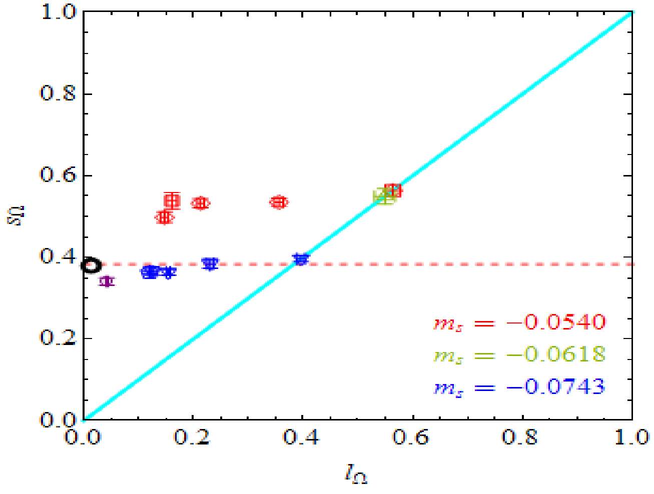
Science Highlight: Excited State Spectrum...

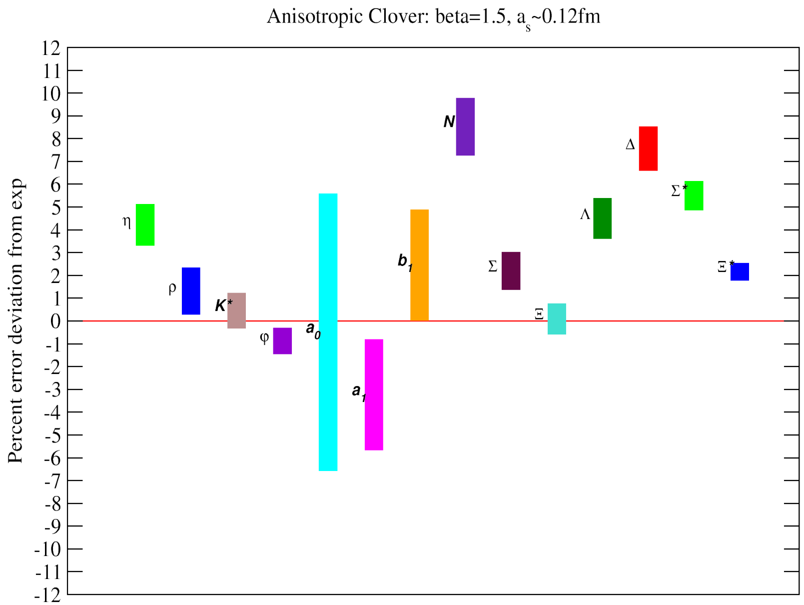


- Excited state spectrum, using anisotropic Wilson Quarks from INCITE'07.
- Successful extraction of some 4 excited states for each group theory channel
- Using degeneracies in channels we identified a (5/2)- state for the first time on the lattice

J. Bulava et. al. Phys. Rev. D 79, 034505 (2009)

Approaching The Physical Quark Mass





- INCITE 2008 & NSF
- found good parameterization of quark masses that lets us
  - determine the physical strange quark mass
  - extrapolate our data to the physical limit
- low lying hadron masses agree with experiment to 10%
  - BMW collaboration is more accurate but for us this is not the main focus
- Working towards excited state spectrum

H-W Lin, et. al. Phys. Rev. D79, 034502 (2009)

Future Work

- Use our INCITE & NSF allocations – obviously
- Produce Physics Results (or we “starve”)
- More optimization – would like better performance if possible.
- Test new technologies (long term)
  - Implement a Wilson Dslash term using UPC?
    - How well do the single ended remote memory accesses work? Will I need Hybrid UPC-MPI mix?
    - Will a UPC Dslash integrate nicely with our existing C++ based code system?
  - Replace expression templates with Domain Specific source transormations (this is a lot of hard work and would need to be done in collaboration with others...)

Conclusions

- There are a lot of beautiful algorithms behind LQCD calculations.
- There is a nice software infrastructure from SciDAC

- All this has produced, and is producing some great physics
  - Excited State Spectra, Hadron Structure, Nuclear Forces

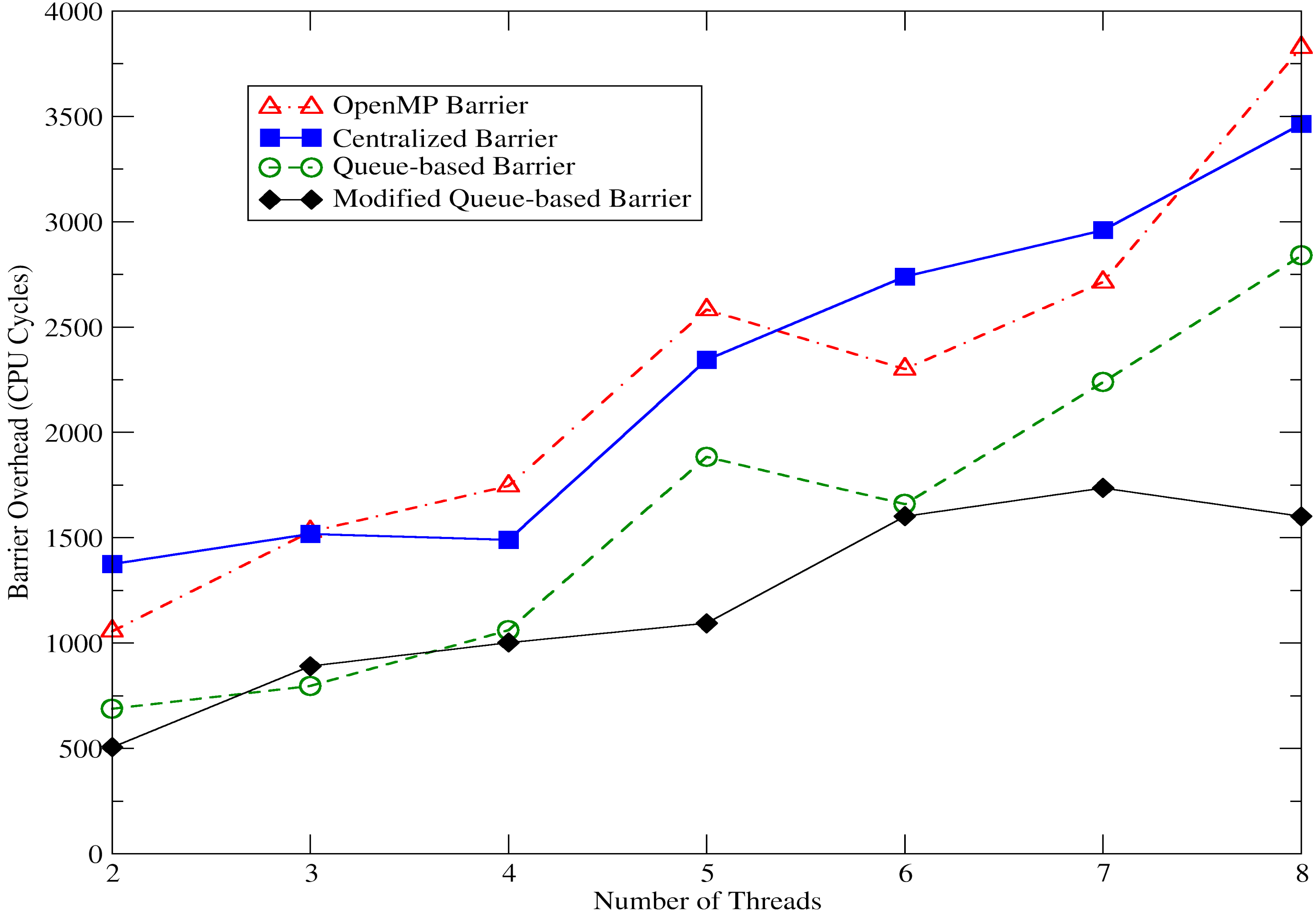
- The performance fluctuations are a little worrying. Resolving these is my highest priority right now. I wish to enlist your continued help for this.
- Finally:

I want to express my thanks to all the staff at NCCS and NICS with whom I had a chance to interact for all their help.

Having access to facilities like Jaguar and Kraken through DOE INCITE and NSF PRAC allocations is awesome. It has enabled our project. We couldn't do it without you.

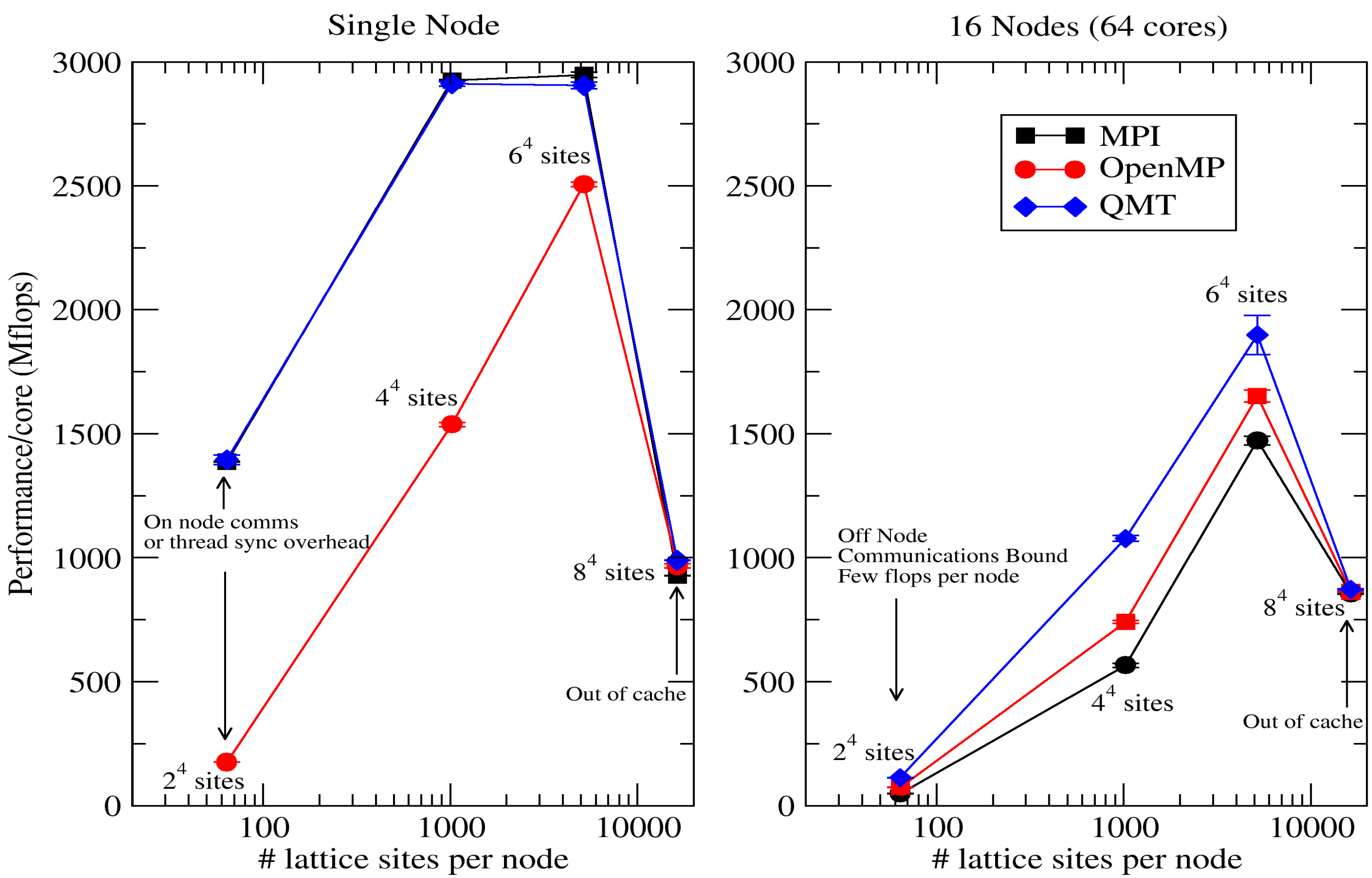
Backup Slides

QMT Microbenchmarks



Initial Multi Threading Tests: Jaguar

4 Threads per node



Summary of Monte Carlo Process

- Importance sampling by Markov Chain Monte Carlo Process
  - HMC: Configurations suggested by Molecular Dynamics
  - MD integrators have to be reversible and area preserving
  - Fermion Forces and Energies require Linear System Solvers
    - Costs increase with decreasing quark mass and a
  - Can keep the process going as long as desired. More configurations reduce statistical errors $\sim 1/\sqrt{N}$ -
- O(10000) traj. runs have very high cost:
  - multi Tflop-year (now) and Pflop-year (future) runs
  - algorithmic improvement is important (deflation, MG)
  - efficiency (as high a performance as we can manage) is important